SmartTokens: Embedding Motion and Grip Sensing in Small Tangible Objects

Mathieu Le Goc, Pierre Dragicevic, Samuel Huron, Jeremy Boy, Jean-Daniel Fekete

SmartTokens: Embedding Motion and Grip Sensing in Small Tangible Objects

Mathieu Le Goc[1,2], **Pierre Dragicevic**[1], **Samuel Huron**[3,4], **Jeremy Boy**[1], **Jean-Daniel Fekete**[1]
[1]Inria, [2]Université Paris-Sud, [3]University of Calgary, [4]IRI Centre Pompidou
mathieu.le-goc@inria.fr, pierre.dragicevic@inria.fr, samuel.huron@cybunk.com, myjyby@gmail.com, jean-daniel.fekete@inria.fr

Figure 1. SmartTokens are small-sized tokens supporting touch and motion sensing, and wireless communication with a coordinator.

ABSTRACT
SmartTokens are small-sized tangible tokens that can sense multiple types of motion, multiple types of touch/grip, and send input events wirelessly as state-machine transitions. By providing an open platform for embedding basic sensing capabilities within small form-factors, SmartTokens extend the design space of tangible user interfaces. We describe the design and implementation of SmartTokens and illustrate how they can be used in practice by introducing a novel TUI design for event notification and personal task management.

Author Keywords
Tangible user interfaces (TUIs); tangible tokens; sensor network user interfaces (SNUIs).

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g., HCI): Miscellaneous

INTRODUCTION
We introduce SmartTokens, small-sized tangible tokens that can sense information on motion and grip and transmit this information wirelessly. SmartTokens can be used as building blocks for developing sensor network user interfaces (SNUIs), i.e., “distributed TUIs in which many small physical manipulatives have sensing, wireless communication” [15]. The unique combination of small form factor and multiple sensing capabilities makes it possible to build token-based SNUIs.

Size constraints offers interaction and manipulation opportunities but introduce several technical challenges, both hardware- and software-related, which we discuss in this paper. We also present a use case of SmartTokens involving notification and task management.

A large amount of previous work on TUIs has focused on tracking the location of tangible objects with external sensing equipment. However, these traditional approaches have limitations. Camera-based tracking systems are sensitive to lighting conditions and are prone to hand occlusion, while tabletop tangibles require tangible objects to rest on a horizontal surface. In addition, the use of external sensing devices restricts the space where tangibles can be operated, and makes it hard to move the user interface from one place to another.

It is possible to build TUIs without the use of external sensing technologies. One solution is to use smartphones as tangible objects (e.g., [13]). Smartphones are widely available but are costly and can be too bulky for some TUI applications, especially when large numbers of tangibles need to be used. More specialized solutions exist like Sifteo cubes [24]. Although these devices are more compact and cheaper than smartphones, some TUIs may ideally require an even smaller form factor, e.g., tokens the size of a two-euro coin that can be grasped easily, either alone or in small number.

To summarize, there is a gap in available TUI technology. Existing devices with embedded sensing capabilities are generally feature-rich but can be costly and bulky. On the other hand, passive props are cost-effective and can be arbitrarily small but they require external sensing equipment. Neither approach is able to fully tap into our ability to manipulate mul-

UIST 2015, November 8–11, 2015, Charlotte, NC, USA.

DOI: http://dx.doi.org/10.1145/2807442.2807488

tiple physical tokens, such as coins or game pieces. To fill this gap, we propose a design that embeds sensing and wireless communication capabilities into small tangible tokens, making them “aware” of their manipulation (see Figure 1).

As such, our main contributions are:

- A small form-factor TUI device that embeds grip and motion sensing at a low cost.
- An open source and open hardware platform to help explore, tinker with, and prototype novel types of TUI designs.
- The “tangible notification machine”, a novel TUI that illustrates how SmartTokens can be employed for the purposes of peripheral notification and personal task management.

BACKGROUND AND MOTIVATION

Tangible objects come in many different form factors. Among them, *tokens* are discrete, usually simple and small physical objects representing a single piece of information [28]. For example, a coin stands for a monetary value, while a piece in a board game can stand for a player. Tokens have been used for thousands of years [23], and more recently, HCI researchers have explored how they can be used to interact with digital information in various applications such as voicemail [2], database queries [27], visualization [8, 9] or games [1]. Tangible tokens can be used as input devices, and occasionally as output devices through self-actuation [20]. Both input and output yield major technological challenges, and in this article we focus on input. Many strategies have been used to support input, and they generally fall in two categories: *external sensing* and *internal sensing*.

External Sensing

External sensing involves the use of sensors that are outside the tangible objects to detect and track information on these objects. A common approach is to use a camera and visual markers to track position and orientation [12, 29]. A related approach is the use of depth cameras [30]. Other spatial tracking technologies include capacitive sensing on touch surfaces [11] and electromagnetic sensing [14]. Finally, RFID tags [19] can be used to detect the nearby presence of tangible objects.

External sensing solutions are able to provide position and sometimes orientation information on tangible objects of arbitrary form factor, including tangible tokens. However, external sensing systems are typically bulky and complex to set up, and often operate under controlled environmental conditions. The interaction space is thus tied to the location of the sensing equipment, which prevents tangible tokens to be seamlessly integrated with our everyday environments [6]. Moreover, external devices cannot easily sense manipulative information such as touch or grip. Therefore, internal sensing has many potential benefits over external sensing for TUIs.

Internal Sensing

Internal sensing refers to systems that embed sensors within the tangible objects themselves. Many types of electronic sensors can be employed, e.g., light, sound, kinetic, accelerometer, gyroscope, and touch sensors. Systems can be classified according to what information they sense and how they communicate this information.

Some systems are entirely self-contained. For instance, “kinetic memory” systems like Curlybot [4] and Topobo [22] are made of physical building blocks that can record and replay manipulations. Another example is Terrenghi et al’s “cube to learn” [26], whose building blocks are screens with acceleration sensors. Although stand-alone tangible systems can support specific applications remarkably well, they cannot be used as input devices to prototype TUIs since they cannot communicate with external systems. Furthermore, we are not aware of any self-contained system that is made of tokens, besides Bishop’s marble answering machine concept [21].

An early attempt to build self-sensing objects with communication capacities was Electronic Blocks [31], a children game to build “computer programs” made out of LEGO bricks with touch, light, and sound sensors, and that can intercommunicate once physically connected. Later Merill et al. [15] coined the concept of *Sensor Network User Interface* and introduced Siftables. Siftables combine a small LCD screen and an accelerometer with wireless communication support. They support two-dimensional topology recognition through contact sensing. Sifteo Cubes [24] are a commercial extension with additional capacitive sensing capabilities. Recently, Hosoi et al. [7] introduced A-Blocks, which support three-dimensional topology recognition through geomagnetism sensors. All these devices have a rather small form factor and rich internal sensing capabilities, but mostly focus on physical assembly scenarios. For example, they do not sense grip.

Bar of Soap [25] is, to our knowledge, the unique attempt to combine grip and motion sensing in a mobile device. The device is able to sense orientation and grip. However, it remains bulky and is meant to be used as a single object.

Besides specialized TUI devices, off-the-shelf generic-purpose mobile devices such as smartphones have also been used to prototype TUIs [13]. Smartphones provide highly elaborate sensing, computation and communication capabilities, but at the cost of price and form factor. Finally, internal and external sensing approaches can be combined. An example of a hybrid strategy is Touchbug [17], a tangible object with embedded accelerometer and gyroscope that detects gestures like shake and tip. Touchbugs are coupled with a multi-touch tabletop for sensing their absolute positions.

Motivation for SmartTokens

As we discussed, many technologies are available to prototype or implement TUIs, each with a different trade-off. A diversity of choices is important for TUI researchers to be able to explore a large array of designs. Internal sensing is receiving more and more attention, and several devices exist that can sense topology, motion, and touch. Yet, form factor remains a bottleneck, as no device exists that has the width of a coin and can support even basic sensing. This limits researchers’ access to the range of TUIs that can be prototyped.

Tokens are ubiquitous in our everyday lives (e.g., monetary coins, bus tokens, game pieces) and their form factor has been refined over thousands of years of technological evolution. Tokens provide an answer to several real-world problems: first, they are easy to transport and easy to store in small containers such as pockets and wallets. They also resist wear and shocks (e.g., when we drop them). Finally, tokens are easy to manipulate (including in-hand) and to hand over, both as individual items and as collections.

Our goal is to provide coin-sized tangible and graspable objects that are *smart* enough to be used as handles for digital information, at a low cost.

SMARTTOKEN DESIGN AND IMPLEMENTATION

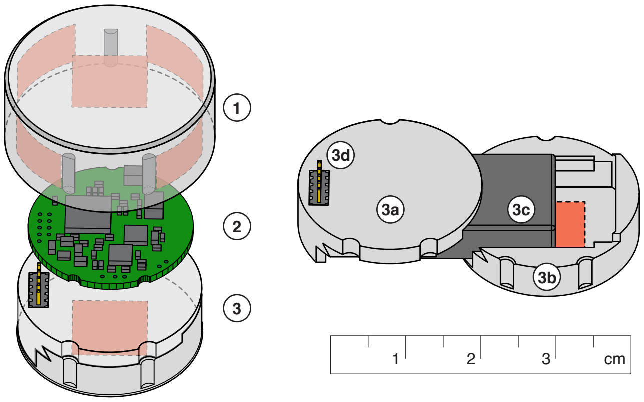

Figure 2. SmartToken Design

In this section we detail the implementation of SmartTokens and the underlying design rationale.

Form Factor and Structure

The SmartToken's shell is a 3D-printed cylinder of 28 mm diameter by 12 mm height (Figure 2 ①). This form factor is comparable to common passive tokens as discussed previously, and is generic enough so it does not carry a specific meaning. A SmartToken is large enough to be easily grasped, while being small enough to allow for group manipulation (about five tokens fit comfortably in a hand). SmartTokens occupy almost half the volume of Siftables [15] and 5 to 6 times less than Sifteo cubes [24] (7.4 cm^3 vs. 13.0 cm^3 vs. 42 cm^3).

All components of SmartTokens fit on a custom circuit board measuring 22 mm in diameter by 2.3 mm high (Figure 2 ②). The electronics design is based on NXP's wireless networking solution [18], using a JN5168 micro-controller. Within each SmartToken, a single micro-controller is in charge of sensing, processing and wireless communication.

Each SmartToken is powered by a 70mAh Lithium-Polymer battery (3c in Figure 2) supplying enough energy for approximatively three hours of intensive use. This battery takes up most of the space. Batteries are recharged using a custom charger, where the battery block (③ in Figure 2) of the SmartToken can be inserted.

SmartTokens are relatively cost-effective: when produced in small quantities, components cost about $20 in total, the circuit board costs about $7 to manufacture, for a total cost of $27 and a total assembly time of about two hours.

Sensing

SmartTokens contain touch sensors and a 3D accelerometer and gyroscope. To ensure a good sensing reactivity, both touch and acceleration sensors are read at a rate of 100Hz. Changes in grip and motion are then used to update an internal state-machine described in the next section.

Touch

A capacitive touch sensor (Atmel AT42QT1070) is embedded in each SmartToken. It monitors six electrodes distributed on the inner faces of the SmartToken's shell (pink areas in Figure 2). To provide the best touch sensitivity, the size of the touch electrodes has to be as large as possible. However, enlarging them excessively would block wireless communication. We found that using rim electrodes of size 10×7 mm (10×10 mm for the upper and lower electrodes) and spacing them apart by 9 mm worked well in our case. Electrodes are connected to the PCB through thin wires.

The touch state of the six electrodes is stored in a byte by the microcontroller and events are delivered to our state-machine when the state of the electrodes changes.

Motion

Motion is monitored with a six axes inertial measurement unit (ST LSM6DS0). Data is delivered as three linear accelerations (angular rates are ignored in the current implementation). Since the accelerometer output includes the gravity (g) component, we estimate g through a calibration phase at startup (while the token is still) and then substract it from all subsequent measurements. Thus the output at rest is 0.

Our state machine distinguishes between immobility, motion and free fall by monitoring changes in acceleration and touch events. *Motion* is inferred when the acceleration exceeds 25 % of g for more than 100 ms, and at least one touch contact is found. When a token falls, the expected output is $|0 - g| = g$. Thus *free fall* is inferred when no touch is registered and acceleration exceeds 80 % of g during at least 20 ms.

Communication

For communication, we use IEEE 802.15.4 ultra low-power wireless communication technology [16]. The network is created and managed by a coordinator embodied by a USB dongle (see Figure 3) and connected to a host computer. The dongle receives state-change messages from all SmartTokens. Each message also contains the state of the touch sensors. On the host computer, a C# server application translates all incoming messages into higher-level events and streams them via a TCP socket to client applications.

Concerning scalability, the IEEE 802.15.4 standard identifies each node with a 16 bits address, so up to 65535 nodes can

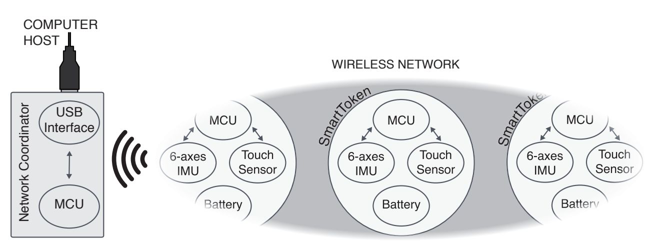

Figure 3. SmartToken Sensor Network Architecture

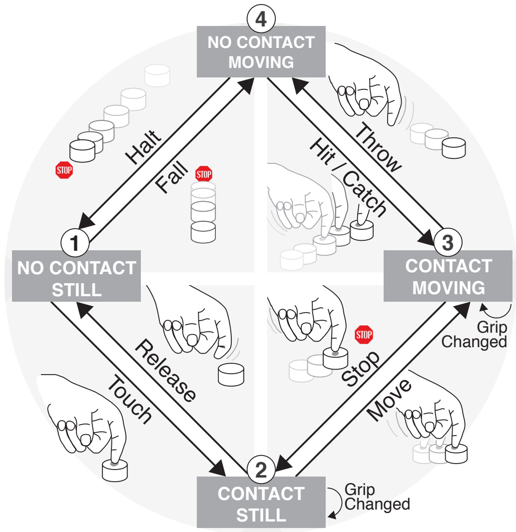

Figure 4. State-machine describing input events a SmartToken can emit.

Figure 5. Four different grips with a SmartToken.

be used at the same time in theory. Raw signals are processed within the SmartTokens themselves, minimizing the amount of information to transmit. A transition is encoded in 4 bytes, for a total of 19 bytes per packet: the used bandwidth is thus much lower than the network's bandwidth (250 kbit/s), allowing multiple SmartTokens to transmit almost simultaneously.

The operating range of an IEEE 802.15.4 network is about 10 meters, which should be enough for most office applications such as our use-case scenario. However, mobile or pervasive computing scenarios would require multiple distributed coordinators, which is possible using the IEEE 802.15.4 standard.

THE SMARTTOKEN INPUT MODEL

Following Buxton [3], we model the input event vocabulary of SmartTokens as a state-machine (see Figure 4). Each SmartToken senses *i)* whether or not it is *touched* by the user and *ii)* whether it is *moving* or still. The cross-product of these two state-machines yields a state-machine with four different states (dark rectangles in Figure 4). Transitions between the states correspond to different manipulative actions (arrows in Figure 4). Eight different transitions are possible, plus four more transitions (not illustrated) if we assume that touch and motion transitions can happen concurrently. Thus monitoring two binary states already provides a rich repertoire of actions. In addition, SmartTokens also sense *grip*, i.e., whether the user touches the object on one side or more (see Figure 5).

USE CASE: TANGIBLE NOTIFICATION MACHINE

To illustrate how SmartTokens can be used in practice, we present an application scenario that consists of a tangible notification and task management interface inspired from Durrell Bishop's marble answering machine [21].

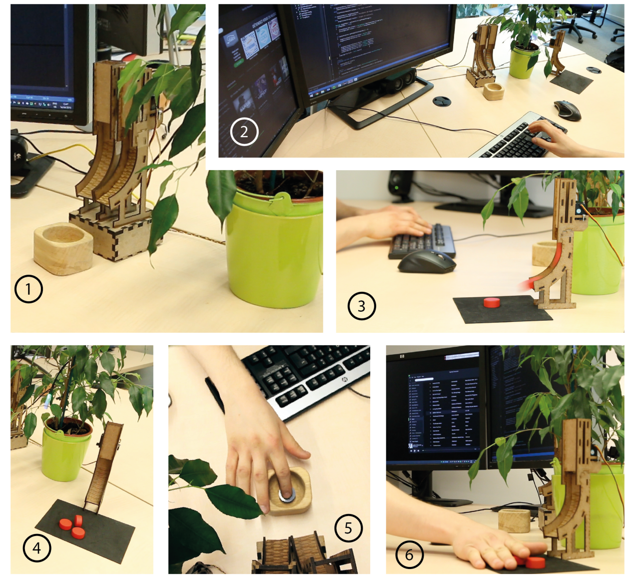

Figure 6. A short strip illustrating the mail notification system.

Walkthrough

The notification machine consists of a set of SmartToken dispensers that communicate with a personal computer (see Figure 6). When an event of interest is received on the computer (e.g., a new calendar event, a bug report or email), a token is released. Different dispensers represent different event types, and SmartTokens accumulate in their respective containers. Tokens can also come in different forms to convey different types of notifications through color, shape or material. SmartTokens can be picked up anytime to display event details on the computer screen or to update an event's status.

For example, suppose Tom is a software engineer who maintains a blog and an open-source software project. He wants to be notified when someone comments on his blog or posts a bug report on his software. He also wants to be notified of any new email, but does not want to be constantly interacting with his email client and prefers to keep his screen free of any distraction. In his fablab he built his own notification machine made of three dispensers, one for each notification type, and bought a kit with SmartTokens of three different colors.

Tom has been coding for three hours, during which three red tokens fell down (see ②–④ in Figure 6). He barely noticed them, as he placed the dispenser out of his sight and a piece of foam absorbs most of the noise of falling tokens. Later a grey token falls in the wooden container (see ① in Figure 6), indicating a blog comment and making a distinctive noise. Out of curiosity he touches it (see ⑤ in Figure 6), and the title of the comment appears on his screen. He then picks up the token and a window pops up showing the entire comment. The comment does not call for a reply, so he drops the token back into the dispenser.

This short interruption prompts Tom to peek at his new emails. He brushes the red tokens still on his desk (see ⑥ in Figure 6), and sees the senders and subjects from all three emails on his screen. One of them is a reminder asking for a prompt reply. He places it in front of his keyboard with a post-it note to make sure he does not forget about it, puts the other back in the dispenser, and gets back to work.

Later, Tom is about to leave when he realizes his email token is still waiting to be dealt with. He pinches it to display the message body again, then decides he will reply from home later tonight. He grabs it and encloses it with his hand while shaking to mark the message as important, and a red star appears next to the message. He puts the token back on the desk and leaves.

Novelty and Benefits

Compared to desktop tools, the notification machine offers a different way of managing notifications and tasks: *i)* since SmartTokens are *tangible*, users can arrange them in a way that is meaningful to them; *ii)* since SmartTokens are capable of *sensing*, they support a range of additional tasks: notification items can be previewed, inspected, and their status changed and reflected back on the computer.

Furthermore, the notification machine provides richer output and input than typical peripheral and ambient devices (e.g., [10]) and offers a different way of interacting with tangible objects compared to classical TUIs. Both the marble answering machine concept [21] and RFID-based TUI systems [19] assign predefined semantics to specific containers or specific spatial locations. In contrast, SmartTokens only care about what happens to *them*, which allows users to assign their own semantics to locations and containers, and leaves more room for the opportunistic use of space and creativity.

Notification Machine Implementation

Assuming the SmartToken framework is provided, implementing the notification machine is relatively effortless and accessible to anyone with basic digital prototyping and programming skills. Each token dispenser is made of wooden laser-cut parts. SmartTokens are stored vertically in the dispenser, and a servomotor arm delivers them one by one. These then fall down a slide. Each dispenser is controlled by a simple Arduino board and connected to the host computer via USB.

The notification manager is a Java program with a GUI to connect notification sources to notification sinks. Each token dispenser is a sink. The driver currently implements a single notification source factory that connects to an email account through the IMAP protocol and lists each folder as a notification source. Many web applications with notification support can be configured to send emails automatically, thus this solution is quite versatile. The library is extensible and can be connected to other notification sources.

The code for associating tokens with notification events and for supporting user interaction with tokens is written on top of *i)* the notification manager framework and *ii)* a Java API for registering to SmartToken input events from the state-machine. The events are used as follows:

Initially, when all SmartTokens are stored in a dispenser, they are in state ① (see Figure 4). When a notification is received and a dispenser releases a SmartToken, the token switches to state ④ and a *fall* event is sent. The notification manager receives this event and maps the ID of the token to the ID of the notification event just received. Then the token moves back to state ① after the fall. As soon as Tom touches it, the token transitions to state ② and emits a *touch* event. The notification's title is then displayed on the screen (email sender and subject) until a *release* event is received. When Tom picks up a token: a touch event is sent, shortly followed by a *grip change* event indicating two contact points; the notification details (email body) is then displayed. In case the token was picked up for the first time (i.e., from a container), the *move* event that follows tags the notification event as *read*. Enclosing the token inside one's hand triggers a *grip change* event on 3 to 6 contact points, which adds the tag *important*. Finally, when Tom drops a token back in its dispenser, the token moves to state ④ as it falls and finally ends in to state ①. The *fall* event removes the notification ID from the token.

CONCLUSION AND FUTURE WORK

We presented SmartTokens, small-sized tangible tokens that can sense multiple types of motion and various types of grip, and can wirelessly send input events modeled as state-machine transitions. SmartTokens extend the design space of TUIs by providing a cost-effective and versatile open-source[1] platform for developing token-based tangible interfaces.

The token form-factor makes it possible to map multiple kinds of digital information, such as notifications or emails, to individual objects that can persist in our physical environment. Users can manipulate up to about 12 of these objects simultaneously; these objects can be stored or used in machines; they can be exchanged between people and carried around (e.g., in one's pocket) across different physical environments. All of these features are difficult or impractical to support with previously proposed TUI devices, which are builkier and meant for different usage scenarios.

We illustrated SmartTokens' capabilities by revisiting Bishop's marble answering machine [21], but the token form-factor is also suitable for a range of other applications, such as board games or physical data storytelling[2]. We are also considering using SmartTokens to facilitate the logging and analysis of interaction with tangible interfaces in empirical studies.

We are currently working on improving the design of SmartTokens. Power consumption is an aspect we did not consider in our first prototype, but day- or even week-long battery life can be achieved using software strategies. For instance, a SmartToken can fall into deep sleep when no activity is detected for a given period of time. Similarly, a reduced sensor sampling rate can be used when the user is not interacting with the SmartTokens. On the hardware side, we are considering using a cheaper and more efficient accelerometer as the gyroscope our current component contains is not used.

[1] http://mathieulegoc.github.io/SmartTokens/

[2] http://dataphys.org/list/tag/storytelling/

SmartTokens can be extended in multiple directions to become even more versatile. Recognizing more input gestures requires no change in our hardware design; we will conduct participatory design sessions to collect gestures worth recognizing. Extending the hardware to provide visual or haptic feedback would also be useful and possible while keeping a similar form-factor. Similarly, being able to locate the tokens would certainly be useful and should be achievable using the same sensing and communication technology [5]. Finally, we plan to adapt the design to other form-factors such as marbles.

ACKNOWLEDGMENTS

We would like to thank Xuemei Yu, Felicia Lin and Can Liu for posing as hand models.

REFERENCES

1. Activision Publishing Inc. Skylanders video game. http://www.skylanders.com/home, 2013.
2. Bishop, D. Marble answer machine. https://vimeo.com/19930744, 1992. Last accessed April 2015.
3. Buxton, W. A three-state model of graphical input. In *INTERACT '90*, vol. 90 (1990), 449–456.
4. Frei, P., Su, V., Mikhak, B., and Ishii, H. Curlybot: Designing a new class of computational toys. In *Proc. CHI '00*, ACM (New York, NY, USA, 2000), 129–136.
5. Goldoni, E., Savioli, A., Risi, M., and Gamba, P. Experimental analysis of rssi-based indoor localization with ieee 802.15.4. In *Wireless Conference (EW), 2010 European* (April 2010), 71–77.
6. Hornecker, E., and Buur, J. Getting a grip on tangible interaction: a framework on physical space and social interaction. In *Proc. CHI '06*, ACM (2006), 437–446.
7. Hosoi, T., Takashima, K., Adachi, T., Itoh, Y., and Kitamura, Y. A-blocks: Recognizing and assessing child building processes during play with toy blocks. In *SIGGRAPH Asia 2014 Emerging Technologies*, ACM (New York, NY, USA, 2014), 1:1–1:2.
8. Huron, S., Carpendale, S., Thudt, A., Tang, A., and Mauerer, M. Constructive Visualization. In *Proc. DIS '14*, ACM, ACM (Vancouver, Canada, June 2014), Pages 433–442.
9. Huron, S., Jansen, Y., and Carpendale, S. Constructing Visual Representations: Investigating the Use of Tangible Tokens. *IEEE TVCG 20*, 12 (Aug. 2014), 1.
10. Jafarinaimi, N., Forlizzi, J., Hurst, A., and Zimmerman, J. Breakaway: an ambient display designed to change human behavior. In *CHI EA '05*, ACM (2005), 1945–1948.
11. Jansen, Y., Dragicevic, P., and Fekete, J.-D. Tangible remote controllers for wall-size displays. In *Proc. CHI '12*, ACM (2012), 2865–2874.
12. Kaltenbrunner, M., and Bencina, R. reactivision: a computer-vision framework for table-based tangible interaction. In *Proc. TEI '07*, ACM (2007), 69–74.
13. Klum, S., Isenberg, P., Langner, R., Fekete, J.-D., and Dachselt, R. Stackables: combining tangibles for faceted browsing. In *Proc. AVI '12*, ACM (2012), 241–248.
14. Liang, R.-H., Cheng, K.-Y., Chan, L., Peng, C.-X., Chen, M. Y., Liang, R.-H., Yang, D.-N., and Chen, B.-Y. Gaussbits: magnetic tangible bits for portable and occlusion-free near-surface interactions. In *Proc. CHI '13*, ACM (2013), 1391–1400.
15. Merrill, D., Kalanithi, J., and Maes, P. Siftables: towards sensor network user interfaces. In *Proc. TEI '07*, ACM (2007), 75–78.
16. Molisch, A. F., Balakrishnan, K., Cassioli, D., Chong, C.-C., Emami, S., Fort, A., Karedal, J., Kunisch, J., Schantz, H., Schuster, U., et al. Ieee 802.15. 4a channel model-final report. *IEEE P802 15*, 04 (2004), 0662.
17. Nowacka, D., Ladha, K., Hammerla, N. Y., Jackson, D., Ladha, C., Rukzio, E., and Olivier, P. Touchbugs: Actuated tangibles on multi-touch tables. In *Proc. CHI '13*, ACM (New York, NY, USA, 2013), 759–762.
18. NXP Semiconductors. Wireless connectivity techzone. http://www.nxp.com/techzones/wireless-connectivity/overview.html, 2015.
19. Paradiso, J. A., Hsiao, K.-y., and Benbasat, A. Tangible music interfaces using passive magnetic tags. In *Proc. NIME '01*, National University of Singapore (2001), 1–4.
20. Pedersen, E. W., and Hornbæk, K. Tangible bots: interaction with active tangibles in tabletop interfaces. In *Proc. CHI '11*, ACM (2011), 2975–2984.
21. Polynor, R. The hand that rocks the cradle. *International Design Magazine* (1995), 60–65.
22. Raffle, H. S., Parkes, A. J., and Ishii, H. Topobo: a constructive assembly system with kinetic memory. In *Proc. CHI '04*, ACM (2004), 647–654.
23. Schmandt-Besserat, D. *How writing came about*. University of Texas Press Austin, 1996.
24. Sifteo cubes. https://www.sifteo.com/cubes/, 2015.
25. Taylor, B. T., and Bove, V. M. The bar of soap: a grasp recognition system implemented in a multi-functional handheld device. In *CHI EA '08*, ACM (2008), 3459–3464.
26. Terrenghi, L., Kranz, M., Holleis, P., and Schmidt, A. A cube to learn: A tangible user interface for the design of a learning appliance. *Personal Ubiquitous Comput. 10*, 2-3 (Jan. 2006), 153–158.
27. Ullmer, B., Ishii, H., and Jacob, R. J. Tangible query interfaces: Physically constrained tokens for manipulating database queries. In *Proc. of INTERACT '03*, vol. 3 (2003), 279–286.
28. Ullmer, B., Ishii, H., and Jacob, R. J. K. Token+constraint systems for tangible interaction with digital information. *ACM Trans. Comput.-Hum. Interact. 12*, 1 (Mar. 2005), 81–118.
29. Weiss, M., Wagner, J., Jansen, Y., Jennings, R., Khoshabeh, R., Hollan, J. D., and Borchers, J. Slap widgets: bridging the gap between virtual and physical controls on tabletops. In *Proc. CHI '09*, ACM (2009), 481–490.
30. Wilson, A. D. Depth-sensing video cameras for 3d tangible tabletop interaction. In *TABLETOP '07*, IEEE (2007), 201–204.
31. Wyeth, P., and Wyeth, G. F. Electronic blocks: Tangible programming elements for preschoolers. In *IFIP TC. 13*, vol. 1, IOC Press (2001), 496–503.